PREPRINT VERSION

Visualizing Queries on Databases of Temporal Histories: New Metaphors and their Evaluation

Luca Chittaro[a] and Carlo Combi[b]

[a]*HCI Lab,*
Department of Mathematics and Computer Science, University of Udine,
via delle Scienze 206, 33100 Udine, Italy
chittaro@dimi.uniud.it

[b]*Department of Computer Science, University of Verona,*
strada Le Grazie 15, 37134 Verona, Italy
combi@sci.univr.it

The published article appeared in Data and Knowledge Engineering, vol. 44, no.2, 2003, pp. 239-264.

Visualizing Queries on Databases of Temporal Histories: New Metaphors and their Evaluation

Luca Chittaro[a] and Carlo Combi[b]

[a]*HCI Lab,*
Department of Mathematics and Computer Science, University of Udine,
via delle Scienze 206, 33100 Udine, Italy
chittaro@dimi.uniud.it

[b]*Department of Computer Science, University of Verona,*
strada Le Grazie 15, 37134 Verona, Italy
combi@sci.univr.it

Abstract

A crucial component for turning any temporal reasoning system into a real-world application that can be adopted by a wide base of users is given by its user interface. After analyzing and discussing the state of the art for the visualization of temporal intervals and relations, this paper proposes three new solutions to the problem of visualizing temporal intervals and their relations for querying databases containing several histories. The metaphors exploited in the proposed visual vocabularies are based on real-world, concrete objects, such as strips, springs, weights, and wires. We discuss the expressivity of the visual vocabularies with respect to the well-known Allen's Interval Algebra. A method for mapping queries composed by the visual vocabularies into SQL queries is then described and discussed. The proposed solutions were evaluated with two proper user studies: the first focused on determining which of the adopted metaphors are more frequently perceived and understood in a correct way and was based on a questionnaire; the second considered the two solutions which scored better in the first phase and studied them with a more thorough experiment, which was also based on user interfaces implementing the two proposals. The visual vocabulary which provided the best results has been adopted in a medical system for visual querying clinical temporal databases.

KEYWORDS: information visualization, temporal query languages, visual queries, temporal data, clinical data

1. Introduction

A crucial component for turning any temporal reasoning system into a real-world application that can be adopted by a wide base of users is given by its user interface. A proper consideration of human-computer interaction (HCI) aspects is needed both in presenting temporal data to users (ensuring that it is easy and quick to understand and does not lead to ambiguous interpretations) and in accepting temporal specifications from them (ensuring an easy formulation of queries or constraints, and minimizing the possibility of errors).

The visual display of temporal information is a recent research direction that has attracted people from diverse fields such as HCI, databases, medical informatics, multimedia, and the new specific field of *Information Visualization* (IV) [2, 4, 6, 8, 10, 12, 14, 16, 17]. Although commonly accepted and used visual representation choices for *intervals* and *timelines* emerge from the literature, much work remains to be done, especially for the representation of *temporal relations* and *complex temporal patterns* for querying databases containing temporal histories, i.e. descriptions of the time-varying properties of some real-world entities.

In this paper, we first briefly survey the state of the art, pointing out some of the open issues we are investigating in our research (Section 2). Then, we sketch a number of requirements for a framework devoted to the visual representation of temporal data (Section 3). Unfortunately, designing an effective visual vocabulary is not a trivial task, because, as pointed out by Chittaro [4], no disciplined design methodologies and engineering principles for information visualization have been yet identified, e.g., temporal relations are *abstract* information which cannot be easily mapped into physical metaphors, because they have no natural and obvious physical representation (therefore, new visual metaphors for representing temporal information are an important research topic). In Section 4, we present our novel proposals for such kind of metaphors, discuss their expressivity, and provide a method for mapping them into database queries. However, to determine if a chosen visualization makes a task easier for a type of user in a given context, it is necessary to carry out proper user studies and evaluations, following the rigorous techniques commonly used in HCI research for studying users and evaluating systems. Therefore, unlike most of the above mentioned approaches, we devote a significant part of the paper (Sections 5 and 6) to evaluate our proposed metaphors and determine what is the best suited for users who are unfamiliar with temporal reasoning concepts, but need to define and enter temporal patterns into a temporal reasoning system. In Section 7, we present the medical application we implemented by exploiting the visual vocabulary which obtained the best results in the evaluation.

2. Background and Motivations

Defining a suitable visual language for temporal data needs the integration of different theoretical as well as methodological work, both from traditional areas devoted to temporal representation (temporal reasoning, temporal databases, temporal logics) and from the recently born area of *Information Visualization* (IV). IV can be defined as "the use of computer-supported, interactive, visual representations of abstract data to amplify cognition" [2]. Temporal information, which is inherently abstract, is thus a proper subject of investigation for IV research, which can greatly contribute to improve user's capability of deeply understanding different facets of temporal information. Indeed, the well-known classification of the different kind of data of interest to IV proposed by Shneiderman [19] explicitly includes temporal data (defined as "data with a start time, finish time, and possible overlaps on a timescale, such as that found in medical records, project management, or video editing"). Although the separation among the different categories is not always strict (in particular, temporal data can be also seen as an instance of multi-dimensional data), Shneiderman underlines the importance of considering it a category in its own to help orienting the choice of IV techniques: when the temporal aspect is dominant in the considered data, display techniques that give a central role to time can give better results than more general techniques which do not assume specific relations among the multiple attributes.

The visualization of temporal information has been considered in different contexts, such as: history representation [16]; display of clinical information [6]; visual query of multimedia video data [12], of relational data [10], and of object-oriented multimedia data [9]; therapy planning [14]; definition of temporal abstractions [17]. One of the first systems proposed for the visualization of temporal data was the Time Line Browser [8] which visualized instant events (such as the measurement of a clinical parameter with its value) and intervals with duration (such as the status of the patient) on a timeline. A more elaborate visualization is proposed in Lifelines [16], the most widely known visualization environment for personal histories. In Lifelines, facts of histories are displayed as lines on a graphic time axis, according to their temporal location and extension; color and thickness are used to represent categories and significance of facts. Visualizing histories on a graphic, linear time axis allows one to temporally compare and relate displayed facts, zooming-in/zooming-out allows one to both overview a whole history and analyze details in more depth (also by selecting items of interest and getting details on demand, e.g., a lab report).

Subsequent systems have basically followed the Lifelines approach, trying to enrich it with further elements. KNAVE [17] is a system that focuses on the visualization and interactive exploration of temporal abstractions (e.g., "hypertension", i.e., high blood pressure) of medical raw data (e.g., a series of measurements of systolic and diastolic blood pressure). Users can dynamically examine temporal information at multiple levels of abstraction (e.g., the measurement of systolic/diastolic blood pressure is more detailed than the hypertension state) and change the level of granularity (e.g., years, days, hours). The AsbruView [14] system for medical therapy plans enriches the timeline visualization by exploiting some 3D elements: besides the two usual dimensions on which the different (possibly overlapping) parts of plans are temporally laid out, a third dimension is used to add graphic elements which convey further information (e.g., when a plan is completed, or might be suspended, or aborted,…). The graphic elements are chosen in such a way that the resulting visualization resembles a running track, which the physician has to run along as the treatment of the patient evolves.

As for database systems, Mquery [9] is a visual query language based on a specific object-oriented data model, which allows one to query multimedia, timeline, and simulation data. Both visual query definition and data visualization is supported by the system, which provides the user with the capability of visualizing time varying (multimedia) data and querying it in a uniform, integrated way. The basic notion of the data model underlying Mquery is the *stream*, which is used to represent any kind of time-varying data. Mquery has been used to manage multimedia temporal clinical databases for thermal ablation therapy and for thoracic oncology.

Every above mentioned approach basically adopts the interval representation proposed in the seminal work of Tufte [20]: a temporal interval is usually displayed by a horizontal bar on a bi-dimensional space, where the x-axis represents the time line and the y-axis is associated to the considered time-varying information. In this way, it is simple to graphically represent any set of intervals if their respective relations are among the 13 classified by Allen [1]: for example, Figure 1 depicts the relation *before* between intervals A and B. A few approaches have recently attempted to propose alternative timeline representations, drawing the timeline along spiral [22] and circular [15] structures, but the practical effectiveness of these less familiar visualizations has not been yet explored with proper user studies.

A relevant problem which is not adequately addressed by current systems is the graphic representation of more complex temporal relations such as temporal patterns. For example, in the medical domain, a physician can need to consider only the histories of those patients who were prescribed aspirin and, after the start of the therapy, had an episode of dyspnea followed by headache. In this case, the focus is not on the visualization of histories, but on the representation of a temporal pattern that can be matched by several histories.

To illustrate the problem in more detail, we consider the two following temporal patterns:

Case 1. Intervals A and B start at the same time, but it is irrelevant when A and B finish.

Case 2. Interval A is equal to interval B; interval C is equal to interval D; the four intervals finish at the same time; A and B start before or together with C and D.

To represent these cases with the visual interval representation described above, one has to resort to disjunctions of Allen's relations; indeed, Case 1 can be modeled by the formula:

(B *starts* A) ∨ (A *starts* B) ∨ (A *equal* B)

which is visually represented in Figure 2, while Case 2 can be modeled by several formulas, e.g.:

(A *equal* B) ∧ (C *equal* D) ∧ ((C *finishes* A)∨ (C *equal* A))

that would also lead to unpractical visual representations. To partially overcome this problem, Hibino and Rundensteiner proposed a specific graphic notation [12]. Figure 3 represents Case 1 with that notation: interval A is represented by a (dark gray) bar; interval B is represented by a segment bounded by two circles (an empty circle for the left end and a filled one for the right end); interval B can terminate in three different positions with respect to the termination of interval A. Unfortunately, this notation cannot deal with more general situations involving more than two intervals, as in Case 2. Moreover, the visualization of the two intervals (one as a bar and one as a segment) and their ends is not based on a single, homogenous graphic choice.

<FIGURE 1 HERE>

<FIGURE 2 HERE>

<FIGURE 3 HERE>

3. Features for a temporal visualization framework

In this section, we introduce and comment on a number of aspects that have to be considered in visualizing temporal information. The general features we identify can be both used to specify requirements for a temporal visualization framework, and to examine and compare existing temporal visualization techniques. First, four basic aspects have to be considered in visualizing temporal information:

- Time points: since the theoretical notion of point has no physical counterpart, time points are usually associated to some graphic objects, as circles, boxes, or ad-hoc icons. Objects are located with reference to a time axis, which is usually represented as an horizontal line. Such line can be left in some cases implicit.
- Time intervals: the usual graphic elements for intervals are boxes or lines; temporal location and extent of intervals are displayed with reference to a (possibly implicit) time axis, as for time points.
- Temporal relations: the relative position among displayed points/intervals is an usual choice for displaying temporal relations. However, this is a solution which does not always address the need of precisely considering relations among points/intervals. Other proposals focus more specifically on the explicit representation of temporal relations, e.g., using labeled arcs.
- Logical expressions: in several situations (e.g., displaying histories), all the represented intervals are (implicitly) related by AND operators, i.e. all the facts associated to the displayed intervals are 'true'. However, in other cases (e.g., querying a database of histories), we want to be able to express disjunctions of relations, as depicted in Figure 2.

<TABLE 1 HERE>

<TABLE 2 HERE>

Further temporal features can be considered:

- Indeterminacy: in real-world information, the temporal location (span) of an interval is often known only with some degree of imprecision. Indeterminacy of temporal points/intervals can bring uncertainty to the associated temporal relations.
- Granularity: temporal information needs to be displayed according to different time units. This allows users to have both an overall view and a detailed view on considered data. A different issue we have to consider is related to the visual representation of different granularities contained in temporal information.

- Temporal views: temporal information can be accessed according to different criteria, graphic notations, user interfaces, which allow the user to focus on different aspects of temporal information: e.g., temporal extents vs. temporal relations.

Tables I and II summarize how the proposals cited in the previous section deal with the above mentioned aspects. In the following, we will focus on the design and the evaluation of metaphors for temporal intervals and their relations.

4. Visualizing Temporal Relations

In this Section, we propose three alternative visual vocabularies for the representation of intervals and their relations which make it easy to visualize and represent temporal queries. The metaphors in each proposal are based on concrete objects and activities from the physical world (e.g., painting and construction) to encourage users to reuse their prior knowledge to develop an understanding of the representation more readily. The metaphors are not inherently related to any application domain (e.g., medicine) to make the visual language useful for any database of temporal histories.

In designing the first proposal, we aimed at keeping the familiar bar representation of temporal intervals and limiting as much as possible the number of new concepts introduced in the traditional interval representation: the interval is graphically represented by a rectangular box in the same way as other approaches do, and the new elements are directly taken from typical physics textbooks' examples representing moving masses. For the other two proposals, we allowed ourselves more freedom in the graphic solutions adopted both for intervals and for the additional elements. In the following, we illustrate each proposal, identifying it with the name of the object it uses to represent a temporal interval.

4.1. Elastic Bands

In the first proposal, intervals are seen as elastic bands. The location of these bands on the time axis can be defined in different ways (see Figure 4, first column):

- Bands' ends can be fixed by screws (example *a* in the figure). In this case, we want to represent intervals' ends that have a precisely set position with respect to other intervals. The commonsense reasoning motivating this choice is that "if the end of a band is fixed by a screw, it cannot move".
- Alternatively, any end of a band can be attached to a moving mass system (inspired by common physics textbooks' figures) as shown in Figure 4 (first column, example *b*). This notation expresses that the end of the interval can take different positions on the time axis: the end of the band can be stretched up to the point it reaches the wall.
- Finally, a moving mass system can be used to connect more than one band simultaneously to represent intervals' ends which can move but keep their relative position. For example, the right ends of the two intervals in Figure 4 (first column, example *c*) can move, but the lower interval will always terminate after the upper one.

Figure 5 (first column) shows how the proposed notation is used to express relations among intervals. In particular, example *a* represents the relation: "interval A is before interval B", while examples *b* and *c* illustrate how to represent the situations respectively described by Case 1 and Case 2 (in Section 2).

<FIGURE 4 HERE>

<FIGURE 5 HERE>

4.2. Springs

The second proposal displays intervals as springs. Any end of a spring can be either fixed with a screw (Figure 4, second column, example *a*) or connected to a weight by means of a wire (Figure 4, second column, example *b*). A weight can be connected to more than one spring simultaneously (Figure 4, second column, example *c*).

The meaning of notations depicted in the second column of Figure 4 is the same of the already described notations in the first column. Analogously, the second column of Figure 5 represents the three examples already described for the first column.

4.3. Paint Strips

The third proposal represents intervals as paint strips. Paint strips can be represented either plainly without any attached object (Figure 4, third column, example *a*) or with a paint roller associated to any of their ends and connected to a weight by means of a wire (Figure 4, third column, example *b*). A weight can be connected to more than one roller simultaneously (Figure 4, third column, example *c*).

The meaning of notations in Figure 4 (third column) and examples in Figure 5 (third column) is the same of that already described for the first and second column.

4.4. Expressivity of the three proposals

The three proposed vocabularies are semantically equivalent, because they differ only in the pictorial elements used to graphically draw intervals and their possible relations. In this Section, we thus analyze their expressivity, by showing what relations can be represented, regardless of the specific pictorial elements. We will illustrate the expressivity of our formalism in terms of the well-known Allen's Interval Algebra (IA) [1], which models the relation between any two intervals as a suitable subset of a set of 13 basic relations, namely, *before, meets, overlaps, starts, during, finishes* (*b, m, o, s, d, f*), together with their inverses (*bi, mi, oi, si, di, fi*) and the *equal* (*e*) relation which coincides with its inverse. The chosen subset of relations for a pair of intervals represents the possible set of relations that might hold between the two intervals. For example, I1 {b, bi} I2 expresses the fact that I1 is *before or after* I2. In this way, 2^{13} different sets of relations can be specified between any pair of intervals.

Our formalism is less expressive than Allen's IA: the sets of relations between any pair of intervals that do not form a *conceptual neighborhood* [11] cannot be specified with our visual vocabularies. We briefly illustrate what a conceptual neighborhood is by recalling two concepts proposed by Freksa in [11] (to which we refer the interested reader for a detailed discussion of conceptual neighborhoods).

First, two relations between pairs of intervals are *conceptual neighbors* if they can be directly transformed into one another by continuously deforming (in a topological sense) the intervals. For example, the relations *b* and *m* are conceptual neighbors, because they can be directly transformed into one another by stretching one of the two intervals (if an interval is *before* another interval, we can stretch any of the two intervals up to the point it *meets* the other one). On the contrary, the relations *before* and *overlaps* are not conceptual neighbors, because a continuous deformation can transform them into one another only indirectly via the relation *meets*.

Second, a set of relations between pairs of intervals forms a *conceptual neighborhood* if its elements are path-connected through 'conceptual neighbor' relations. For example, {b, m, o} forms a conceptual neighborhood because the relations *before*, *meets*, and *overlaps* can be transformed into one another by a chain of direct continuous deformations of the associated intervals (if an interval is *before* another interval, we can stretch any of the two intervals up to the point it *meets* the other one, and then we can stretch it further until the two intervals are in a *overlap* relation). On the contrary, {b, o} does not form a conceptual neighborhood (and is thus an example of a relation that can be specified in IA, but not in our approach).

Although they represent only a subset of IA, conceptual neighborhoods are very relevant because they represent those sets of relations which are perceived to be natural from a cognitive point of view, as discussed in general by [11]. Our practical experience with clinicians strengthens Freksa's claims. The design of our visual vocabularies was indeed based on discussions and case analyses performed with some clinicians, and only when we later decided to formally study the obtained vocabularies with respect to the temporal reasoning literature, we noticed the strict relation with Freksa's work.

Since the conceptual neighborhoods one can obtain are dependent on the choice of topological deformation (e.g., one could move intervals instead of stretching them), it is important to highlight which kind of deformation we support. As the reader can easily note, some elements of our visual vocabularies (moving mass system for Elastic Bands, wire and weight for Springs; paint roller with weight for Paint Strips) are devoted to intuitively represent the idea of continuous deformation by *stretching*. As a result, the conceptual neighborhoods determined in this way are among those called *A-neighborhoods* in Freksa's terminology [11]. We now provide the reader with a procedure for deriving all possible IA relations between a pair of intervals that can be visually specified through our formalism. When using our approach, the user must first draw the two intervals in a initial position (the single basic relation expressed by this initial position can be any of the 13 Allen's basic relation). Then, the user can define possible continuous deformations of each of the two intervals, by specifying which ends of the intervals can be stretched and how much. To determine what is the resulting set of relations, we distinguish three possible user actions:

1) The user applies stretching elements of our vocabularies on just one end of only one of the two intervals. Tables 3 and 4 show every possible set of relations that can be specified in this way. Both tables include also the degenerate cases where the length of the stretching is null (this is equivalent to not applying at all the stretching element, keeping only the basic relation specified by the initial position).
2) The user applies stretching elements to both ends of only one of the two intervals. The resulting relations are obtained by union of the possible relations already given in the two tables: for example, considering A {d} B as the initial position, by stretching A both to the right and to the left (see the ninth line of Tables 3 and 4) we can obtain any relation that is the union of one element of the set {{d}, {d,f}, {d,f,oi}} with one element of the set {{d}, {d,s}, {d,s,o}}.
3) The user applies stretching elements to both intervals. The resulting relations can be obtained as follows: first, the relations resulting from stretching only one of the intervals are obtained (see the two previously discussed cases). Then, each obtained basic relation is considered as a possible initial position for the two intervals, and the stretching of the second interval is applied to that initial position, obtaining the remaining relations. For example, consider the situation in Figure 6, where M {o} L (or, equivalently, L {oi} M) is the initial position of two intervals, and the user has applied two stretching elements. By considering only the stretching of M, we obtain M {o, fi} L (or, equivalently, L {oi, f} M). Now, for each of the two possible positions of L with respect to M, we consider the effect of stretching L, obtaining respectively {oi, si, di}, and {f,e,fi}. The union of these possibilities gives the full relation specified by the user, i.e. L {oi, si, di, f, e, fi}M.

<TABLE 3 HERE>

When a constraint (see Figure 4, line c) is applied to the ends of the two considered intervals, the derivation of the IA relations between the two intervals is simplified, because there is no need to consider the possible relations that can arise from relative changes in the considered ends: for example, in the cases depicted in Figure 4 (line c), the side to which the constraint has been applied can be disregarded.

<FIGURE 6 HERE>

<TABLE 4 HERE>

4.5. Mapping visual patterns into SQL queries

The visual temporal patterns are composed, as previously outlined, for query purposes: temporal facts stored into the database are matched against the given pattern in order to select specific situations which are significant to the user. In other words, the visual temporal pattern specified by the user represents a query on a database.

In this section, we will show how the visual patterns are translated into queries on the underlying database. Without loss of generality, we will use a generic SQL as the target query language. We consider queries on a *history database*, which contains a set of *temporal histories*; each temporal history is composed by one or more *temporal facts*. Let us now introduce in a more formal way the concepts of history database and of temporal history, respectively.

Definition 1. A *history database* is a relation on the schema *HistoryDB(HistoryID, FactDescr, FactStart, FactEnd)*.

Definition 2. A *temporal history* is any subset of a history database that contains all the tuples of the history database having the same value for the attribute *HistoryID*. Each tuple of a temporal history represents a *temporal fact*.

The attribute *HistoryID* allows one to identify a specific history. For example, in the clinical domain, *HistoryID* can identify the patient the clinical history refers to, or the single hospitalization described by that history. The attribute *FactDescr* contains the description of specific facts which compose a history; *FactStart* and *FactEnd* represent the lower and upper ends of the interval during which the considered fact is true in the application domain (in other words, by *FactStart* and *FactEnd* we represent the valid time [13] of the fact). Unlike the previous section, we will use here a point-based approach: this is mainly due to the fact that usually temporal intervals are not supported by SQL and database management systems, while it is possible to represent time points by the DATE and TIMESTAMP types.

For the sake of simplicity, we will mention only the pictorial elements of Paint Strips, but the translation process between visual patterns and SQL queries remains identical for the corresponding pictorial elements of the other two visual vocabularies. With respect to a given history database, a paint strip represents any fact: colors and labels of the paint strip allow one to associate to the paint strip only facts which have specific values for the attribute *FactDescr*.

A visual pattern is formally represented by identifying the relative temporal location of the ends of paint strips. To this purpose, each end *e* of a paint strip is represented by the quintuple *(LowB, UppB, StripID, FactDescr, StartOrEnd)*, where *LowB* and *UppB* contain the values of the lower and upper bounds for the temporal location of *e* on the timeline, which is implicitly represented as an horizontal line where time advances from left to right (*LowB* and *UppB* are different only when *e* is visually represented as a paint roller connected to a weight); *FactDescr* contains the description of the specific fact, visually identified by the label and the color of the paint strip; *StripID* identifies the considered paint strip (with a label related to the vertical position of the strip; see the numbers near strips in Figure 7);

StartOrEnd specifies whether *e* represents the start or the end of the paint strip and thus the *FactStart* or *FactEnd* of the corresponding temporal fact.

In the following, by [*r.a*] we mean the value of the attribute *a* of the relation *r*, returned as a string. We will use the addition (+) symbol for concatenating strings. Symbols “ ” will be used to explicitly contain a string. Let us assume that a query returns the values of the attribute *HistoryID* for those histories in the database which satisfy the given visual pattern. Thus, the SQL query corresponding to the visual temporal pattern will have the following structure:

```
SELECT HistoryID
FROM <source>
WHERE <join_condition> AND < facts_selection> AND < temporal_selection>
```

The <*source*> part can be easily defined: it will contain the single relation *HistoryDB* and a tuple variable on this relation for each paint strip. We can use the value of *StripID* as the name of the corresponding tuple variable.

To build the selection condition of the WHERE clause, we distinguish three different parts: <*join_condition*> contains the join condition that allows one to consider facts belonging to single histories; <*facts_selection*> contains the conditions on the atemporal properties of facts represented by paint strips; <*temporal_selection*> contains the temporal relations among facts as specified by the overall visual pattern. More precisely, <*join_condition*> is built by using the *StripID*s of strips: each tuple variable on the relation *HistoryDB* must have the same value for the attribute *HistoryID* (i.e., the considered temporal facts must belong to the same history); <*facts_selection*> is composed by the conjunction of the equalities obtained by considering, for each strip, the *FactDescr* of any end *e*: [*e.StripID*] + “`.FactDescr =`” + “`"`” + [*e.FactDescr*] + “`"`”. Building <*temporal_selection*> requires more care. A simple solution could be inspired to the approach taken in the previous section: for each pair of paint strips, we can generate the disjunction of the allowed IA relations, and then express them as relations between interval endpoints. The final <*temporal_selection*> condition can then be obtained as the conjunction of all the obtained disjunctions. Unfortunately, this simple approach would fail to provide a selection condition without redundancies: as an example, since equality would be imposed for each pair of ends that have the same temporal location, several superfluous equalities would be evaluated by the database system to execute the query. Therefore, query evaluation can become highly inefficient.

The problem of redundancy in the selection condition can be avoided by taking into consideration the transitivity property for equality and precedence between time points in the design of the translation process. To illustrate how we derive <*temporal_selection*>, we introduce some definitions to classify ends in a pattern with respect to their positions. Let *m* be the number of paint strips in the temporal pattern (there is thus a total of *2m* ends). Let *v* be the number of different temporal values specified by the *LowB* and *UppB* attributes of the *2m* ends in the pattern: there will be at most $n=2*2m=4m$ values, and $2 \leq v \leq n$. The considered v values are in a strict temporal order of v positions, and the function *tpos(i)* will be used to return the temporal location of the ith position. For each position i ($1 \leq i \leq v$), we identify interesting ends which involve that position, by defining the following 3 sets (which have empty intersection):

$FixedAt_i = \{ e \mid e.LowB = e.UppB = tpos(i)\}$

$UpTo_i = \{ e \mid e.LowB \neq e.UppB \wedge e.UppB = tpos(i)\}$

$From_i = \{ e \mid e.LowB \neq e.UppB \wedge e.LowB = tpos(i)\}$

The set $FixedAt_i$ contains the fixed ends (i.e., ends without roller and weight) at position i, the other two sets contain the variable ends (i.e. ends with roller and weight) whose range of possible locations respectively stops ($UpTo_i$) or starts ($From_i$) at position i.

We build <*temporal_selection*> as the conjunction of 8 subformulas, respectively obtained by considering the precedence relations: a) among fixed ends that have different positions; b) between fixed ends and the temporally following variable ends; c) between variable ends and the temporally following fixed ends; d) between variable ends and the temporally following variable ends; e) between variable ends and fixed ends that share the same lower bound; f) between variable ends and fixed ends that share the same upper bound; g) among fixed ends that share the same position (this is done to impose equalities); h) given by explicit constraints between different ends. Subformula h) needs to be generated when more paint strips are connected to a single weight; in these situations, the problem is indeed split in two parts: first, we consider these strips as if they were connected to single different weights, and we handle them in subformulas b), c), d), e), and f); then, we explicitly represent the constraints between these strips specifying the proper relation (equality or precedence) in subformula h). To show in detail how we proceed in the derivation of each subformula, let us consider as an example subformula b), which we obtain as follows:

```
b:= "";
ForEach i,j: 1 ≤ i < j ≤ v ∧ FixedAt_i ≠ ∅ ∧ From_j ≠ ∅ ∧ ¬∃k (i < k < j ∧ FixedAt_k ≠ ∅)
        ForEach e ∈ From_j
                If (FixedAt_i– {e1 | e1.stripID = e.stripID}) ≠ ∅ then
                        ex := any (FixedAt_i– {e1 | e1.stripID = e.stripID})
                        If b ≠ "" then
                            b:= b + " AND "
                        Endif
                        b:= b + [ex.StripID]+"."+[ex.StartOrEnd] + "<" + [e.StripID]+"."+[e.StartOrEnd]
                EndIf
        EndFor
 EndFor
```

where the function any() simply returns one random object from the set given as argument. The derived subformula imposes the precedence relation only between one of those fixed ends which immediately precede a variable end (i.e., there are no other fixed ends in between) and the variable end itself: precedence relations between that variable end and other possible fixed ends will hold by transitivity from relations in other subformulas. As an example, let us consider the visual temporal query represented in Figure 7. It is translated according to the previous approach as:

<FIGURE 7 HERE>

```
SELECT HistoryID
FROM HistoryDB h3, HistoryDB h5, HistoryDB h8
WHERE h3.HistoryID = h5.HistoryID AND h3.HistoryID = h8.HistoryID AND
    h3.FactDescr = "antidepressants" AND h5.FactDescr = "corticosteroids" AND
    h8.FactDescr = "analgesics" AND h8.FactStart < h3.FactStart AND
    h3.FactStart < h5.FactStart AND h5.FactStart < h8.FactEnd AND
    h3.FactEnd < h5.FactEnd AND h5.FactStart < h3.FactEnd AND
    h3.FactEnd < h8.FactEnd
```

In this query, we can easily identify the different parts we have previously outlined:

- <*source*> is, in this case, `HistoryDB h3, HistoryDB h5, HistoryDB h8`, where tuple variables are built by considering the *StripID* appearing near each strip in Figure 7.
- <*join_condition*> is composed by `h3.HistoryID = h5.HistoryID AND h3.HistoryID = h8.HistoryID`.
- <*facts_selection*> consists of `h3.FactDescr = "antidepressants" AND h5.FactDescr = "corticosteroids" AND h8.FactDescr = "analgesics"`.
- <*temporal_selection*> is `h8.FactStart < h3.FactStart AND h3.FactStart < h5.FactStart AND h5.FactStart < h8.FactEnd AND h5.FactStart < h3.FactEnd AND h3.FactEnd < h8.FactEnd AND h3.FactEnd < h5.FactEnd`. In this case, subformula b) consists of `h5.FactStart < h3.FactEnd`, and the condition `h8.FactStart < h3.FactEnd` is not included, since it can be derived by transitivity, for example, from the formula `h8.FactStart < h3.FactStart` AND `h3.FactStart < h5.FactStart AND h5.FactStart < h3.FactEnd`.

In general, it is interesting to note that the temporal relations represented by our visual patterns translate into conjunctions of relations between time points that employ three possible operators (i.e., $<$, $=$, $<=$). As a consequence, those temporal relations belong to those that can be expressed by Continuous Endpoint Algebra (CEA) [21], a well-known proper subset of Allen's Interval Algebra [1].

5. Evaluation: first experiment

Since the three proposed approaches are semantically equivalent, the first phase of our evaluation focused on determining which of the adopted metaphors are more frequently perceived and understood in a correct way and was based on a questionnaire. This section describes how we carried out the evaluation and the obtained results.

5.1. The questionnaire

The evaluation was based on a questionnaire, containing four exercises for each of the three proposals (i.e., a total of 12 exercises), and organized in two different parts:

1. The purpose of the first part (one exercise for each proposal) is to assess which objects are correctly perceived by users as having some freedom of movement. Subjects are shown the situation in line *b* of Figure 4. For each of the three shown examples, a multiple choice question asks subjects to identify which objects can move. For example, considering objects in the figure for the Elastic Bands proposal, the correct answer should indicate only the elastic band, the mass and the spring as possibly moving objects. Figure 8 shows the question for this example.

2. The purpose of the *second part* (three exercises for each proposal) is to assess how much the possible temporal locations and respective temporal relations of intervals are correctly perceived. For each of the three proposals, subjects are presented with each of the three situations illustrated in the corresponding column of Figure 4. For each situation, subjects are asked to precisely state the position of every intervals' end for both the minimum and maximum interval extension. For example, Figure 9 shows the eight positions the user is asked to specify for a situation which corresponds to example *c* in Figure 4 for the Paint Strips proposal. As an example of correct answers, the smallest extension for the green strip (i.e., the upper strip) goes from point 1 to point 4, while its largest extension goes from point 1 to point 5.

<FIGURE 8 HERE>

<FIGURE 9 HERE>

5.2. Experiment Design and Procedure

The questionnaire was administered to 30 subjects (13 females and 17 males). Age ranged from 24 to 37, averaging at 27. Nine subjects were physicians, while 17 were university students (1 in Mathematics, 2 in Computer Science, 3 in Engineering, 8 in Business Administration, 3 in Agricultural Sciences), 3 subjects recently completed their Master (2 in Philosophy, 1 in Biology), and 1 subject held a secretarial position. The majority of subjects (19 people) had taken a course in Physics in their university curriculum (and were thus expected to be familiar with the moving mass notation adopted in the Elastic Band proposal).

Subjects were not given any information about the meaning of the specific graphic elements in the three proposals. Each subject was first asked to fill the first part of the questionnaire, then (s)he was provided with the second. Since each part contained exercises for all three proposals, the order in which they were presented was changed for each subject to minimize learning effects. In particular: (i) every proposal was presented an approximately equal number of times as first, second, and third in both parts of the questionnaire, (ii) the order of presentation of the three proposals in the two parts was different for the same subject, and (iii) the order of presentation of the three exercises given for each of the three proposals in the second part was different for the same subject.

5.3. Analysis and Results

For each subject, we counted how many of the exercises were solved correctly. We applied the most restrictive requirements for correctness: in the first part of the questionnaire, exercises were considered correctly answered if all and only the possibly moving objects were identified; in the second part, an exercise was considered correctly answered only if every required position was indicated correctly. Figures 10, 11, and 12 show the frequencies of the number of correctly answered exercises for each of the three proposals.

Statistical analysis has been performed by applying the Friedman non-parametric test for dependent samples. The within-subjects variable was the type of visual vocabulary with three levels. The dependent variable was the number of correctly answered exercises. The result of the test ($\chi_r^2=10.4$, $p<0.01$) indicated that the effect was significant. We thus employed the multiple comparisons procedure suggested in [7] for post-hoc analysis. The values of means are 2.2 for the Elastic Bands condition, 2.8 for the Springs condition, and 3 for the Paint Strips condition. Post-hoc comparison of means pointed out that the correctness results obtained with Elastic Bands are significantly lower than those obtained with Springs (statistical significance $p<0.05$) and significantly lower than those obtained with Paint Strips ($p<0.01$), while the difference between Springs and Paint Strips turns out not to be statistically significant.

<FIGURE 10 HERE>

<FIGURE 11HERE>

<FIGURE 12 HERE>

6. Evaluation: second experiment

On the basis of the results obtained from the first experiment, we decided to carry out a more thorough evaluation of those two proposals which gave better results than Elastic Bands but did not show statistically significant differences between them. Since the two graphic notations were similar, we did not expect large differences in user performance to come out from this second phase; however, we wanted to acquire at least some sound additional elements of decision to choose which approach to adopt. For this second experiment, we also introduced Visual C++ implementations of the user interface. Figures 7 and 13 show screenshots of the implemented Paint Strips interface and Springs interface, respectively. Interaction with both interfaces was analogous to common drawing applications: the user can create and position the available graphic objects in the window and then set their desired size by using only the mouse (numeric labeling of the intervals is performed automatically by the interface). The only difference between the two interfaces was the available set of graphic objects.

6.1. Experimental Task

We chose to perform this evaluation in a medical context for three reasons. First, as previously pointed out, temporal data and patterns are of particular relevance for applications in the medical domain. Second, we are involved in the design of a medical system for visual querying clinical temporal databases [5]. Third, clinicians are a very good example of users who are neither expert in querying databases nor willing to spend much time learning complex query languages.

The experimental task consisted of two parts. In the first part, subjects were asked to solve 4 *interpretation exercises* on paper. Each exercise showed a temporal pattern, and proposed 3 possible interpretations to choose from, of which only one was correct. In the second part of the task (*definition exercises*), the subject used the graphic interface to visually define 4 different temporal patterns which were described (only in natural language) on a sheet given to him/her.

All the temporal queries used for the experimental task concerned medical situations (e.g., “retrieve those patients who started a therapy with both antibiotics and antidepressants, in the case where it is irrelevant which drug was suspended first”), and were of a complexity analogous or slightly higher than the query depicted in Figures 7 and 13.

6.2. Experiment Design and Procedure

Subjects were recruited at the Medical Clinic of our University. None of them had been involved in the first experiment described in the previous section. A total of 31 people (15 males and 16 females) were recruited: 6 students in Medicine, 4 medical doctors (MDs) who had just earned their degree, 18 MDs specializing in various subfields of medicine (2 in Clinical Pharmacology, 10 in Public Health, 5 in Clinical Psychiatry, 1 in Surgery), 1 psychologist, 1 pharmacologist, and 1 physician employed in the Public Health Department. Age ranged from 23 to 44, averaging at 30. With respect to computer usage, only one subject never used a computer; the others were equally split among those that use it for only a couple of hours per week and those who use it for 5 or more hours per week.

Each subject performed the task in two different sessions (one for each interface). Each session began with a training phase, where subjects were shown the interface and told about the meaning of its graphic elements. During training, the subject was first guided to directly interact with the interface: for each graphic object, (s)he learned how to insert, modify, and delete it. Then, the meaning of each graphic object in the context of temporal patterns was explained. The subject was invited to ask for any clarifications (s)he wanted during the training phase, because it was not possible to do it in the subsequent parts of the session.

<FIGURE 13 HERE>

When subjects felt ready, they were introduced to the experimental task, and then they carried it out. After task completion, the user filled a 28-questions user's satisfaction questionnaire inspired to the QUIS (Questionnaire for User Interaction Satisfaction) [3].

We did not impose any temporal constraint on the duration of the different parts of the session. Since each session lasted about 50 minutes, the two sessions for the same subject were scheduled in different days to avoid excessive tiredness. Since most of the doctors involved in the experiment had very busy schedules, 50 US$ were paid for each subject that participated in the test. To minimize learning effects on the experiment results, different users took the Springs session and the Paint Strips session in opposite orders.

In each session, we collected the following quantitative data: time spent to complete the interpretation exercises, number of correct answers to interpretation exercises, time spent to complete the definition exercises, and number of correct answers to interpretation exercises. Qualitative impressions were recorded with the user's satisfaction questionnaire. At the end of the second session, a written question also asked which of the two used interfaces was best. Subjects were finally verbally asked to explain their choice of the best interface.

The hardware used for the experiment was a standard 17 inch monitor and a Pentium III PC.

6.3. Analysis and Results

The number of correct answers to exercises were analyzed using the Wilcoxon test for two related samples. For the interpretation exercises, the average number of correct answers was 3.39 for Springs and 3.52 for Paint Strips; for definition exercises, it was 2.55 for Springs and 2.81 for Paint Strips. However, both results failed to meet the 0.05 threshold for p.

Data concerning time spent to complete the two parts of the task were analyzed using the paired-samples *t* test. Subjects spent on average less time to solve interpretation exercises in the Paint Strips condition (160 sec.) than in the Springs condition (189 sec.), Paint Strips scored (slightly) better also with definition exercises (567 sec. for Paint Strips and 595 sec. for Springs). Both results failed to meet the threshold for statistical significance, but the first result was close to significance (t = 1.94, p=0.061). In each answer to the user's satisfaction questionnaire, Paint Strips scored slightly better than Springs, except for one question concerning the easiness of perceiving the possibility of movement of intervals. However, the most significant result came from the final question which asked to choose the best interface: 22 of the 31 subjects chose Paint Strips. Significance of this result has been computed with Pearson's Chi-Square test for one-way tables (χ_r^2=6.25, p<0.025).

<FIGURE 14 HERE>

7. Medical application based on Paint Strips

We employed Paint Strips in a system for the visual definition and use of temporal clinical abstractions. A *temporal abstraction* provides a concise and high-level description of a collection of time-stamped raw data [18]. In medical informatics, temporal abstraction plays a central role in supplying care providers with data at a suitable level for supporting decision making. Temporal abstraction on clinical data has thus been investigated in some depth in recent years. In this context, Shahar and Musen [18] proposed a general framework for abstraction of time-stamped data, called the Knowledge-Based Temporal-Abstraction (KBTA) Method. The output of KBTA includes the basic abstractions of type *state*, *gradient*, *rate* (e.g., LOW, DECREASING, and FAST are some abstractions of the three types for values of the hemoglobin clinical parameter) and a special type of abstraction (*pattern*), defined in terms of patterns of basic abstractions (e.g., HYPERGLYCEMIA overlaps GLYCOSURIA ABSENT in monitoring diabetic patients [18]). The KBTA method has been implemented in the RÉSUMÉ system and evaluated in several clinical domains [18].

In our system, we focused on the definition of composite abstractions (which can be defined as *patterns* or *patterns of patterns* in the KBTA terminology), starting from basic abstractions and facts, e.g. simple symptoms or the administration of drugs. The user interface of the system is organized into two windows: the upper one is devoted to the visual specification of temporal patterns in terms of Paint Strips and to their association to a composite abstraction (definition of the name and validity interval of the new abstraction); the lower window shows the results of querying the patient history database (containing the basic clinical abstractions and facts concerning different patients), using the visually defined composite abstraction. Figure 14 is a screenshot of the user interface after the definition of an abstraction (called “SuspiciousPattern”) involving two components. In the upper window, named “Temporal Abstraction Editor”, the user can perform several actions: insertion, modification, deletion, and connection of different graphic objects can be performed by switching among the different options either in the “Temporal” menu and its sub-items or through the suitable toolbar shown in the figure. In the lower window, named “TAV” (Temporal Abstractions Visualizer), the results of the query display abstractions as boxes, following the approach of Lifelines [16], on an implicit horizontal time line. As shown in Figure 14, it is possible to display only the composite abstraction (see the interval visualized for patient 1) or the abstraction together with its components (see the intervals visualized for patient 2), by clicking on the patients’ list on the left part of the window. Different toolbars, scrollbars, and tooltips allow the user to zoom in and out the temporal dimension and the vertical dimension of boxes, to show/hide patients abstractions and/or their components, and to display some reference on the timeline (e.g., the day January 4, 2000, in Figure 14).

8. Discussion and Conclusions

In this paper, we have proposed, analyzed, and evaluated three new approaches to the visual definition of queries on temporal histories. We have proposed visual vocabularies which exploit metaphors based on real world, concrete objects, such as strips, springs, weights, and wires. The expressivity of the visual vocabularies has been illustrated in terms of the well-known Allen’s Interval Algebra: the possible set of relations between any pair of intervals that do not form a *conceptual neighborhood* cannot be represented with our visual vocabularies. Moreover, a method for mapping visually defined temporal patterns into SQL queries has been proposed, also avoiding redundancies in composing the selection condition.

As for the evaluation phase, although we did not expect large differences in the capability of users to correctly understand the metaphors used in the three proposals, the first experiment we carried out showed that the Elastic Bands proposal (which was potentially more familiar to users) gave significantly lower results than both the other two proposals. Some factors which could explain this lower performance of users with Elastic Bands are: (i) the graphic element representing the elastic band might not be immediately perceived as elastic, and (ii) although the adopted graphic representation of a moving mass system is common, it does not graphically represent the possible presence of an external force with a concrete object, making it less evident that the interval can be extended (on the contrary, the Springs and the Paint Strips proposals made this aspect more evident by adopting the less common, but more evident picture of a weight).

In the second experiment, while quantitative data showed only very slight differences in user performance with the two interfaces, qualitative data indicated a significant user preference for Paint Strips. Verbal comments from subjects motivated this result with different answers. The most frequent fall in three categories: Paint Strips allow for a simpler identification of interval endpoints (5 subjects); Paint Strips allow for a more direct interpretation of the situation (5 subjects); Paint Strips are more satisfactory from a visual point of view (4 subjects).

Following the design guidance provided by the described evaluation, Paint Strips have been adopted in the medical system we designed and implemented [5].

The research directions we are currently pursuing concern: (i) the addition and handling of constraints on the duration of single intervals (or distances between separate intervals), (ii) the extension of the proposed visual vocabularies to deal with temporal indeterminacy and different time granularities, (iii) the handling of the conjunction, disjunction, and negation of visual temporal patterns to compose more complex queries.

Acknowledgements

This work is partially supported by a MURST COFIN 2000 project ("Analysis, Information Visualization, and Visual Query in Databases for Clinical Monitoring"). We are grateful to Claudia Vicenzotto for her work in carrying out the experiments with subjects.

References

[1] J. Allen, Maintaining knowledge about temporal intervals, Communications of the ACM 26 (1983) 832-843.

[2] S. Card, J. MacKinlay and B. Shneiderman, Readings in Information Visualization: Using Vision to Think (Morgan Kaufmann Publishers, San Francisco, 1999).

[3] J.P. Chin, V.A. Diehl, K.L. Norman, Development of an instrument measuring user satisfaction of the human-computer interface, in: Proc. of the CHI '88 Conference on Human Factors in Computing Systems (ACM Press, New York, 1988) 213-218.

[4] L. Chittaro, Information Visualization and its Application to Medicine, Artificial Intelligence in Medicine Journal 22 (2001) 81-88.

[5] L. Chittaro and C. Combi, Visual Definition of Temporal Clinical Abstractions: A User Interface based on Novel Metaphors, in: Proc. of the AIME '01 Artificial Intelligence in Medicine Conference, Lecture Notes in Computer Science, Vol. 2101 (Springer, Berlin, 2001) 227-230.

[6] C. Combi, L. Portoni and F. Pinciroli, Visualizing Temporal Clinical Data on the WWW, in: Proc. AIMDM'99 Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making, Lecture Notes in Artificial Intelligence, Vol. 1620 (Springer, Berlin, 1999) 301-311.

[7] W.J. Conover, Practical Nonparametric Statistics, 2nd Edition (Wiley and Sons, New York, 1980).

[8] S.B. Cousins and G. Kahn, The visual display of temporal information, Artificial Intelligence in Medicine 3 (1991) 341-357.

[9] J.D.N. Dionisio and A.F. Cardenas, MQuery: A Visual Query Language for Multimedia, Timeline and Simulation Data, Journal of Visual Languages and Computing 7 (1996) 377-401.

[10] S. Fernandes Silva, U. Schiel, and T. Catarci, Visual Query Operators for Temporal Databases, in: Proc. TIME '97 International Workshop on Temporal Representation and Reasoning (IEEE Computer Press, Los Alamitos, CA, 1997) 46-53.

[11] C. Freksa, Temporal reasoning based on semi-intervals, Artificial Intelligence 54 (1992) 199-227.

[12] S. Hibino and E.A. Rundensteiner, User Interface Evaluation of a Direct Manipulation Temporal Visual Query Language, in: Proc. of the ACM Conference on Multimedia (ACM Press, New York, 1997) 99-107.

[13] C.S. Jensen and R.T. Snodgrass, Temporal Data Management, IEEE Transactions on Knowledge and Data Engineering 11 (1999) 36-44.

[14] R. Kosara and S. Miksch, Metaphors of Movement: A Visualization and User Interface for Time-Oriented, Skeletal Plans, Artificial Intelligence in Medicine 22 (2001) 111-131.

[15] K. Muniandy, Visualizing time related events for intrusion detection, in: Late Breaking Hot Topics Proceedings, InfoVis 2001 Symposium on Information Visualization, 2001.

[16] C. Plaisant, B. Milash, A. Rose, S. Widoff, and B. Shneiderman, LifeLines: Visualizing Personal Histories, in: Proc. of the CHI ’96 Conference on Human Factors in Computing Systems (ACM Press, New York, 1996) 221-227.

[17] Y. Shahar and C. Cheng, Model-Based Visualization of Temporal Abstractions, Computational Intelligence 16 (2000) 279-306.

[18] Y. Shahar and M.A. Musen, Knowledge-based temporal abstraction in clinical domains, Artificial Intelligence in Medicine 8 (1996) 267-298.

[19] B. Shneiderman, The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations, in: Proc. of the IEEE Symposium on Visual Languages (IEEE Computer Society Press, Los Alamitos, CA, 1996) 336-343.

[20] E.R. Tufte, The Visual Display of Quantitative Information (Graphics Press, Cheshire, CT, 1982).

[21] P. van Beek and R. Cohen, Exact and approximate reasoning about temporal relations, Computational Intelligence 6 (1990) 132-144.

[22] M. Weber, M. Alexa and W. Mueller, Visualizing Time-Series on Spirals, in: Proc. of the IEEE InfoVis Symposium on Information Visualization (IEEE Press, Los Alamitos, CA, 2001) 7-14.

CAPTIONS:

Figure 1. Displaying relation “A before B”.

Figure 2. Displaying relations for Case 1.

Figure 3. Displaying Case 1, according to the notation proposed in [12].

Figure 4. Graphic notation for the different proposals.

Figure 5. Three examples of interval relations represented with the different proposals.

Figure 6. An example of two intervals, both with stretching elements.

Figure 7. A visual query with the Paint Strips Interface.

Figure 8. An exercise in the first part of the questionnaire (text has been translated into English).

Figure 9. An exercise in the second part of the questionnaire (text has been translated into English).

Figure 10. Correct answers for Elastic Bands.

Figure 11. Correct answers for Springs.

Figure 12. Correct answers for Paint Strips.

Figure 13. A screenshot of the Springs Interface.

Figure 14. Medical Application based on Paint Strips.

Table 1. Different approaches in visualizing time points, intervals, relations, and logical expressions.

Table 2. Different approaches in visualizing indeterminacy, granularity and temporal views.

Table 3. Stretching only a right end of one interval.

Table 4. Stretching only a left end of one interval.

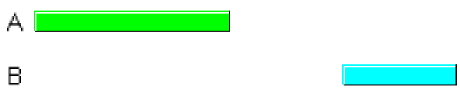


Figure 1. Displaying relation "A before B".

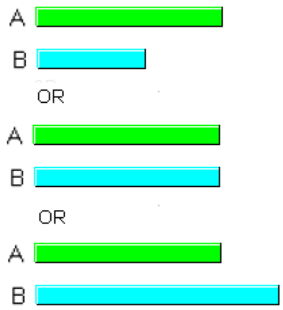


Figure 2. Displaying relations for Case 1.

Figure 3. Displaying Case 1, according to the notation proposed in [12].

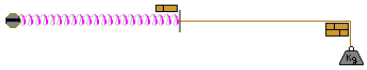
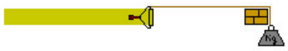
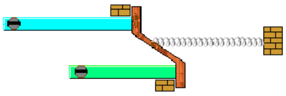
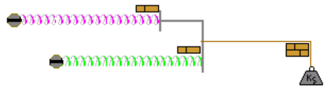
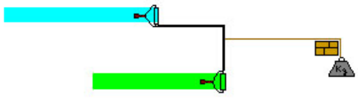
	Elastic Bands	Springs	Paint Strips
(a)			
(b)			
(c)			

Figure 4. Graphic notation for the different proposals.

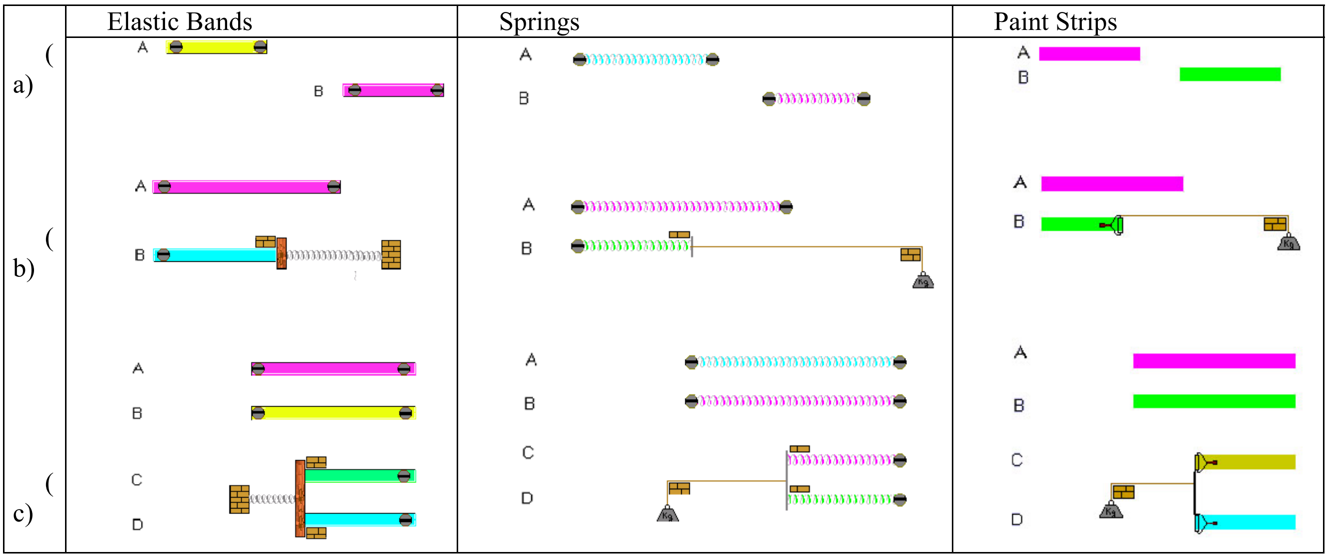


Figure 5. Three examples of interval relations represented with the different proposals.

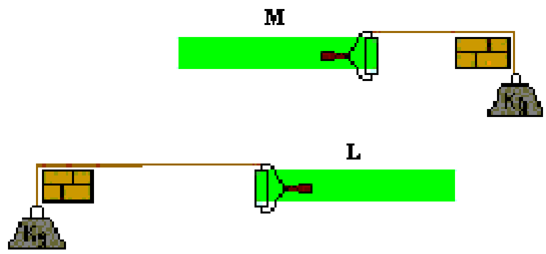


Figure 6. An example of two intervals, both with stretching elements.

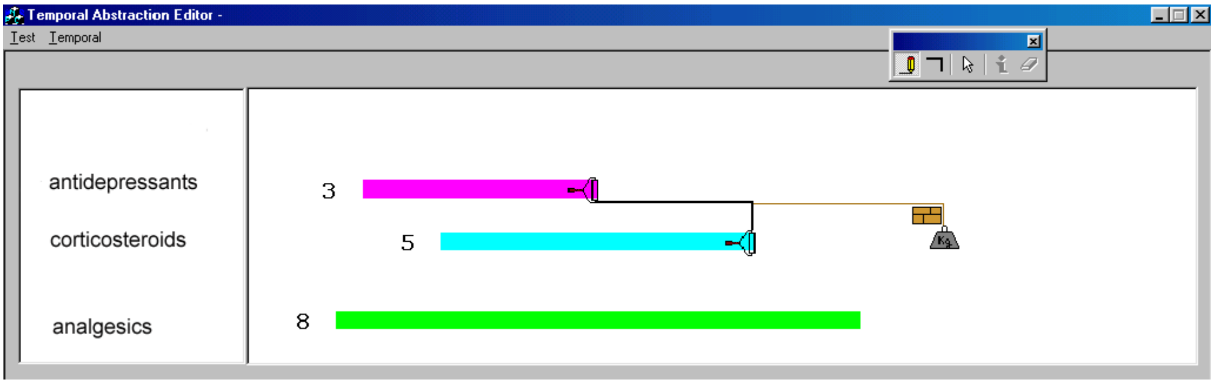


Figure 7. A visual query with the Paint Strips Interface.

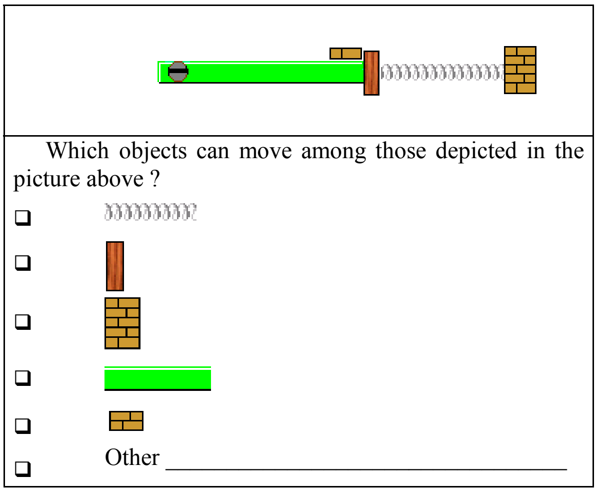
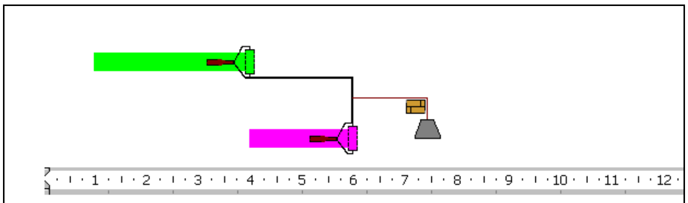


Figure 8. An exercise in the first part of the questionnaire (text has been translated into English).



	Smallest extension		Largest extension	
	From point	To point	From point	To point
Green strip				
Purple strip				

Figure 9. An exercise in the second part of the questionnaire (text has been translated into English).

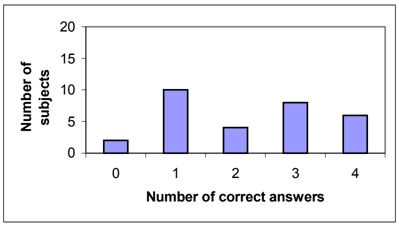


Figure 10. Correct answers for Elastic Bands.

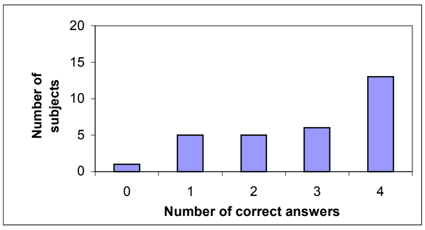


Figure 11. Correct answers for Springs.

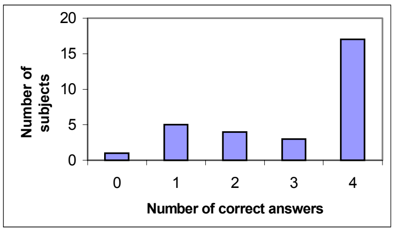


Figure 12. Correct answers for Paint Strips.

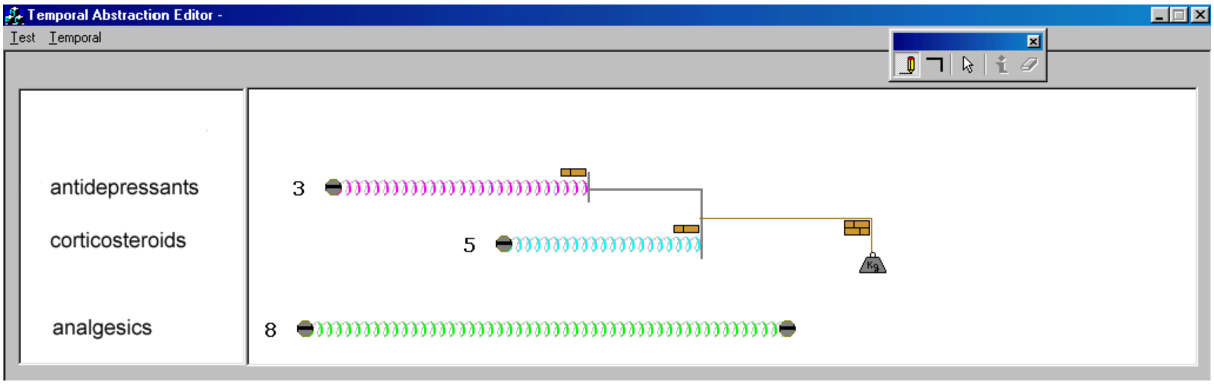


Figure 13. A screenshot of the Springs Interface.

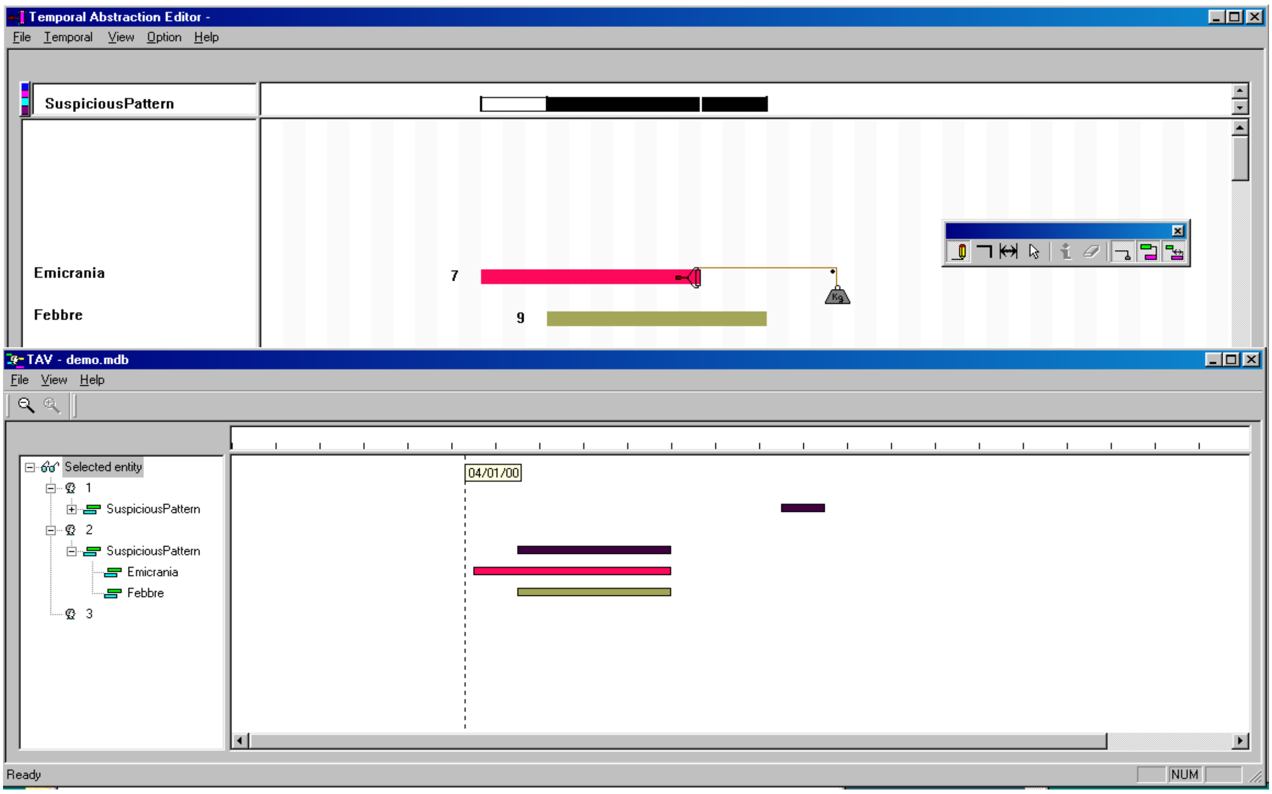


Figure 14. Medical Application based on Paint Strips.

	Time point	Time interval	Temporal relations	Logical expressions
Time Line Browser [8]	circles	boxes	derived from the position of circles and boxes on the time axis	(implicit) conjunction of all the visualized facts
TVQL [12]	not considered	sliders, boxes, lines	determined with 4 sliders	neighbor relations and disjunction of relations
LifeLines [16]	icons	colored boxes with different height	derived from the position on the time axis of boxes/icons	(implicit) conjunction of all the visualized facts
TVQO [10]	boxes	boxes	determined with 2 sliders	not considered
KNAVE [17]	labeled circles	labeled/colored boxes	derived from the position on the time axis of boxes/circles	(implicit) conjunction of all the visualized facts
KHOSPAD [6]	labeled, colored boxes or circles	labeled, colored boxes or circles	labeled and colored arcs	(implicit) conjunction of all the visualized facts
AsbruView [14]	not considered	colored boxes/ running tracks	derived from the position of boxes/tracks on the time axis	(implicit) conjunction of all the visualized facts (some can be optional)
Mquery [9]	boxes	boxes	textually entered	user-defined

Table 1. Different approaches in visualizing time points, intervals, relations, and logical expressions.

	Indeterminacy	Granularity	Temporal views
Time Line Browser [8]	not considered	different granularities can be displayed on the time axis	different icons for different data types
TVQL [12]	not considered	fixed	not applicable
LifeLines [16]	not considered	different granularities can be displayed on the time axis	use of different subareas of the displayed window, with different colors, boxes, and icons
TVQO [10]	not considered	interactively set by the user	not applicable
KNAVE [17]	not considered	interactively set by the user	use of different subareas of the displayed window, with different colors, boxes, and icons
KHOSPAD [6]	indeterminacy of starting, ending points and duration of intervals, relations uncertainty	interactively set by the user	two different views: history oriented and relation oriented
AsbruView [14]	grey color of indeterminate durations, circles and zigzag for interval endpoints	interactively set by the user	two different views
Mquery [9]	not considered	interactively set by the user	user-defined

Table 2. Different approaches in visualizing indeterminacy, granularity and temporal views.

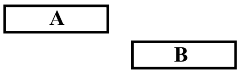
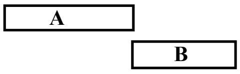
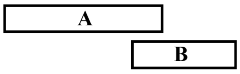
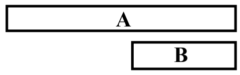
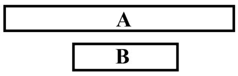
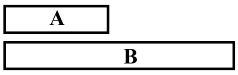
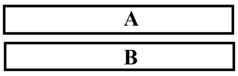
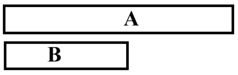
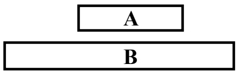
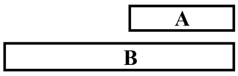
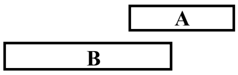
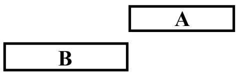
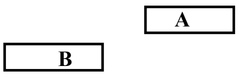
Initial Position	Possible relations between A and B specified by stretching ONLY the RIGHT end of A
	{b}, {b,m}, {b,m,o}, {b,m,o,fi}, {b,m,o,fi,di}
	{m}, {m,o}, {m,o,fi}, {m,o,fi,di}
	{o}, {o,fi}, {o,fi,di}
	{fi}, {fi,di}
	{di}
	{s}, {s,e}, {s,e,si}
	{e}, {e,si}
	{si}
	{d}, {d,f}, {d,f,oi}
	{f}, {f,oi}
	{oi}
	{mi}
	{bi}

Table 3. Stretching only a right end of one interval.

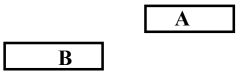
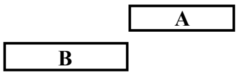
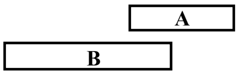
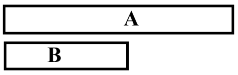
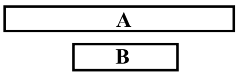
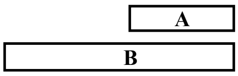
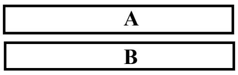
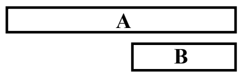
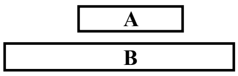
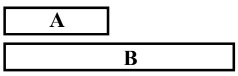
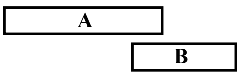
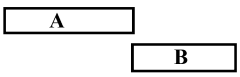
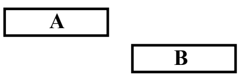
Initial Position	Possible relations between A and B specified by stretching ONLY the LEFT end of A
	{bi},{bi,mi},{bi,mi,oi},{bi,mi,oi,si},{bi,mi,oi,si,di}
	{mi},{mi,oi},{mi,oi,si},{mi,oi,si,di}
	{oi},{oi,si},{oi,si,di}
	{si},{si,di}
	{di}
	{f}, {f,e}, {f,e,fi}
	{e}, {e,fi}
	{fi}
	{d}, {d,s}, {d,s,o}
	{s}, {s,o}
	{o}
	{m}
	{b}

Table 4. Stretching only a left end of one interval.